**How to get current date and time in java**

[**JAVA DATE**](http://beginnersbook.com/category/technology/java-guide/java-date/)

Using SimpleDateFormat and Date/Calendar class, we can easily get**current date and time** in Java. Below are the code snippets of both the ways:

**Current date and time can be obtained using two methods:**

1) **Using Date class**

* Specify the desired pattern while creating object of SimpleDateFormat.
* Create an object of Date class.
* Call the [**format() method of DateFormat class**](http://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#format(java.util.Date)) and pass the date object as a parameter to the method.

DateFormat df = new SimpleDateFormat("dd/MM/yy HH:mm:ss");

Date dateobj = new Date();

System.out.println(df.format(dateobj));

2) **Using Calendar class**

* Specify the desired date/time. Similar to the step 1 of above method.
* Create an object of Calendar class by calling getInstance() method of it.
* Call the format() method of DateFormat and pass theCalendar.getTime() as a parameter to the method.

DateFormat df = new SimpleDateFormat("dd/MM/yy HH:mm:ss");

Calendar calobj = Calendar.getInstance();

System.out.println(df.format(calobj.getTime()));

**Complete java code for getting current date and time strong**

import java.util.Date;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Calendar;

public class GettingCurrentDate {

public static void main(String[] args) {

//getting current date and time using Date class

DateFormat df = new SimpleDateFormat("dd/MM/yy HH:mm:ss");

Date dateobj = new Date();

System.out.println(df.format(dateobj));

/\*getting current date time using calendar class

\* An Alternative of above\*/

Calendar calobj = Calendar.getInstance();

System.out.println(df.format(calobj.getTime()));

}

}

Output:

02/01/14 20:21:16

02/01/14 20:21:16

Every time I run the above code it would fetch the current date and time.

Note: In order to get the output in above format I have specified the date/time pattern above (Note the first statement of the programDateFormat df = new SimpleDateFormat("dd/MM/yy HH:mm:ss"); you would get the output like:02-01-2014

However if you want the output in any other [**date format**](http://beginnersbook.com/2013/04/java-date-format/) just modify the pattern accordingly. For e.g. To get the date only in dd-MM-yyyy format: replace the statement with this one: DateFormat df = new SimpleDateFormat("dd-MM-yyyy");

While specifying the pattern be careful with the case. For e.g. ‘s’ (small s) represents second while ‘S'(Capital s) represents Millisecond.

Here is the complete chart which will help you to define the pattern inSimpleDateFormat.

|  |  |  |  |
| --- | --- | --- | --- |
| **Letter** | **Date or Time Component** | **Presentation** | **Examples** |
| G | Era designator | Text | AD |
| y | Year | Year | 1996; 96 |
| Y | Week year | Year | 2009; 09 |
| M | Month in year | Month | July; Jul; 07 |
| w | Week in year | Number | 27 |
| W | Week in month | Number | 2 |
| D | Day in year | Number | 189 |
| d | Day in month | Number | 10 |
| F | Day of week in month | Number | 2 |
| E | Day name in week | Text | Tuesday; Tue |
| u | Day number of week (1 = Monday, …, 7 = Sunday) | Number | 1 |
| a | Am/pm marker | Text | PM |
| H | Hour in day (0-23) | Number | 0 |
| k | Hour in day (1-24) | Number | 24 |
| K | Hour in am/pm (0-11) | Number | 0 |
| h | Hour in am/pm (1-12) | Number | 12 |
| m | Minute in hour | Number | 30 |
| s | Second in minute | Number | 55 |
| S | Millisecond | Number | 978 |
| z | Time zone | General time zone | Pacific Standard Time; PST; GMT-08:00 |
| Z | Time zone | RFC 822 time zone | -0800 |
| X | Time zone | ISO 8601 time zone | -08; -0800; -08:00 |